

リアルタイム 3DCG における写実性向上の一検討

5 N-1 2

川瀬 正樹 (中部大学), 高橋 友一 (中部大学)

1 はじめに

近年, 安価で高性能なグラフィックスハードウェアが次々に開発されており, 従来, 計算負荷のため困難であった写実性を重視した CG が可能になってきている。

OpenGL など, 一般的なリアルタイム 3DCG API では, 物質表面の素材表現に不可欠なハイライトの表現に, フォンのモデル (Phong Model)[1] と呼ばれる光源モデルが, またスムーズシェーディング [2] として, グローシェーディング (Gouraud Shading) が採用されている。

しかしこれらの手法では, 磁器, プラスティックや車のボディなど, 非常に光沢のある素材を表現する際, 以下のような問題が生じる。

1. フォンのモデルで扱う光源情報がその方向ベクトルおよび色情報 (R,G,B 輝度) だけであるため, 鏡面反射によるハイライト表現に光源の形状を反映できない。
2. グローシェーディングは頂点単位に光源計算を行い, ポリゴン内部の色は頂点色からの線形補間で決定するため, ポリゴンより小さいサイズのハイライトが破綻する。
3. ポリゴンに寄与する全ての光源を独立して計算する必要があるため, 光源数に比例して計算負荷が増大する。

2. については, ポリゴンを細分化することである程度緩和できるが, 根本的な解決にはならない。グローシェーディングではなくフォンシェーディング (Phong Shading) を用いるとこの問題は解消できるが, 計算負荷が非常に高く, リアルタイム処理においては実用的ではない。また, 1. や 3. の解決にはならない。

そこで, このような問題を解決し, リアルタイム 3DCG で写実性を向上させる手法を検討した。また, 実際に OpenGL を用いて描画処理を行い, 従来の手法と速度面, 画質面で比較した。

Improving Computer Graphics Realism using Environment Mapping.

Masaki Kawase, Tomoichi Takahashi

Chubu University

1200 Matumoto, Kasugai-shi, Aichi, 487-8501

2 環境マッピングを用いたハイライト表現

ハイライト表現を, 以下の手順で処理する。まず拡散反射のみを考慮した光源計算結果で, 物体表面の模様テクスチャマッピングを行いながら描画する。次に, 全く同じ頂点情報で予めハイライトを描き込んでおいた環境マップを使用し, 前の描画の上から加算描画する。

この手法による描画では, 以下のような特徴がある。

1. 環境マッピング時のテクスチャ座標は, ポリゴン表面で反射した視線ベクトルから計算する。ピクセル単位に環境マップを参照するため, フォンシェーディングに近い効果を期待できる。
2. 光源の向きや色だけでなく, 光源形状などの情報も含めて作成しておくことができるため, 独特の形状を持った光源をハイライトへ直接反映させることができる。
3. ハイライト描画については, 光源数を増やしても計算負荷は増えない。

3 評価実験

OpenGL を使用し, 約 2400 ポリゴンで構成したオブジェクトを手法 1 (拡散反射シェーディングとフォンのモデルにテクスチャマッピングを施す手法) と手法 2 (拡散反射シェーディングのみでテクスチャマッピングし, その上から環境マッピングによるハイライトを加算描画する手法) でそれぞれ描画し, 写実性及び速度の評価を行った。

3.1 写実性の評価

写実性の評価は数値的な判断が困難であるため, アンケート調査を行った。

マウスで操作可能なオブジェクトを手法 1 と手法 2 で各々 10 秒ずつ描画し, CG に関して予備知識のない学生 10 人を対象に, どちらの方が写実性が高いと感じたかを直感で答えてもらった。また, ディスプレイ装置による差を考慮し, NEC MultiSync と, 富士通の FMV-DP97X2 の 2 種類のディスプレイで比較した (表 1)。

結果は, FMV-DP97X2 では 10 人中 9 人, NEC MultiSync では 10 人全員から, 手法 2 の方が写実

表 1: 写実性評価実験

FMV-DP97X2		NEC MultiSync	
手法1	手法2	手法1	手法2
1人	9人	0人	10人

性が高いとの回答を得た。

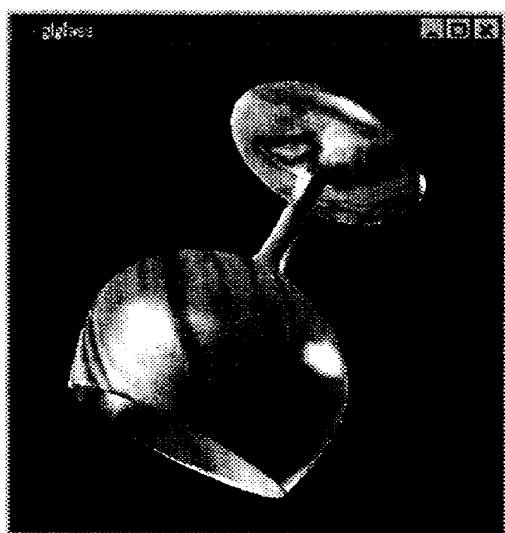


図 1: 手法 1 による描画

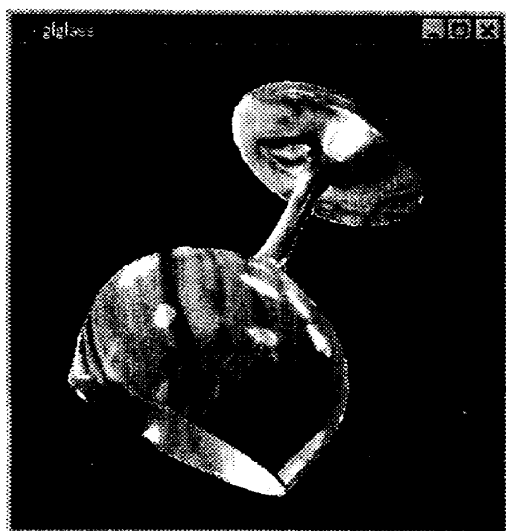


図 2: 手法 2 による描画

手法 1 ではハイライト部分でポリゴンの境界が目立っており、ポリゴンで構成されている事が分かる (図 1)。これに対し手法 2 では、ポリゴン同士の境界は認識できず、ポリゴンで構成していることもほとんど分からない。また、ハイライトに光源の形

状が反映している (図 2)。

3.2 速度の評価

オブジェクトを一秒間に何枚描画できたかを測定し、その平均値で評価した。また、処理能力の異なる CPU (Pentium 133MHz, MMX Pentium 200MHz, PentiumII 300MHz) で比較し、さらにハードウェアアクセラレーションが可能な環境では、ソフトウェアによる描画とハードウェアによる描画の両方で測定した。ただし描画ウィンドウのサイズは、一般的なアプリケーションで必要と思われるサイズ (640 × 480 ピクセル) とした (表 2)。

表 2: 速度評価実験

	Pentium 133MHz	MMX 200MHz		PentiumII 300MHz	
	Soft	Soft	Hard	Soft	Hard
手法 1	0.41	0.90	63.0	1.69	109.2
手法 2	0.19	0.38	36.0	0.70	57.2
手法 1/2	0.46	0.42	0.57	0.41	0.52

表の最下段は、手法 2 の手法 1 に対する速度比である。OpenGL で手法 2 の描画を行う場合、全く同じ頂点変換を 2 回行うため、必要以上の速度低下を招いている。ただ、ハードウェアによる描画では、ソフトウェア程の速度低下は起こっていない。

4 今後の展望

DirectX6[3] の新機能や OpenGL-1.2[4] の新しい拡張機能など、主要な API で採用されつつあるマルチテクスチャ機能 (2 種類以上のテクスチャを一度に指定し、混合しながらマッピングする機能) を使用すると、2 段階に分けている処理を 1 回にまとめることができ、頂点変換の無駄をなくすることができる。

今後、マルチテクスチャ機能にハードウェアで対応したビデオチップが増加することは十分に予想できるため、将来的に実用性の遥かな向上が期待できる。

参考文献

- [1] 山岡 祥:
Turbo C による 3D グラフィックス. 森山出版, 1995.
- [2] Andrew S. Glassner:
3D COMPUTER GRAPHICS. アスキー出版局, 1991.
- [3] <http://www.microsoft.com/directx/default.asp>.
- [4] <http://www.sgi.com/Technology/OpenGL/>.